

Applicant(s): Frank Oliver Hoffmann, *et al.*
U.S.S.N.: 10/816,445

Attorney Docket No.: 34874-096/2003P00878US

RECEIVED
CENTRAL FAX CENTER

JAN 02 2008

REMARKS

Applicant amended independent claim 1 to clarify that it is the defined message is an application message, to further clarify that the application message has a structured application header comprising information relating to one or more component with each component relating to a corresponding set of attributes of the message, and to clarify that the structured header is defined in accordance with an application messaging protocol. Support for the clarification is provided throughout the application, including, for example, at page 5, paragraph 25 and page 6, paragraph 28. Applicant similarly amended independent claims 5, 9, 10 and 11. Applicant also amended independent claim 5 to correct an antecedent problem. After entering the above amendments, claims 1-12 will be pending, with claims 1, 5, 9, 10, and 11 being independent.

The examiner rejected claim 9 under 35 U.S.C. § 102(b) for allegedly being anticipated by U.S. Publication No. US2002/0184357 to Traversat et al. The examiner rejected independent claim 1 under 35 U.S.C. § 103(a) as being unpatentable over Traversat in view of U.S. Publication No. US2002/0138618 to Szabo, and rejected claims 2-8, 10-12 under 35 U.S.C. § 103(a) as being unpatentable over Traversat in view of Szabo, and further in view of U.S. Publication No. US2003/0014733 to Ringseth et al. THESE REJECTIONS ARE RESPECTFULLY TRAVERSED

Applicant's independent claim 1 recites "defining an application message having a structured application message header, the structured message header being defined in accordance with an application messaging protocol, the structured message header comprising one or more components defined by the protocol with each of the one or more components relating to a corresponding set of attributes of the message". Thus, messages include a structured application message header, defined by an application messaging protocol (i.e., a protocol based upon which application messages can be formed) that includes one or more components relating to respective corresponding message attributes (e.g., description of the message content, reliable message component attributes, security component attributes, etc.)

Applicant contends that none of the references relied upon by the examiner discloses or suggests at least the features "defining an application message having a structured application message header, the structured message header being defined in accordance with an application messaging protocol, the structured message header comprising one or more components defined by the protocol with each of the one or more components relating to a corresponding set of attributes of the message".

Applicant(s): Frank Oliver Hoffmann, *et al.*
 U.S.S.N.: 10/816,445

Attorney Docket No.: 34874-096/2003P00878US

Specifically, Traversat describes peer-to-peer network computing platforms (Traversat, page 1, paragraph 7). Traversat explains:

[0147] In one embodiment, the peer-to-peer platform may use asynchronous messages as a basis for providing Internet-scalable peer-to-peer communication. The information transmitted using pipes may be packaged as messages. Messages define an envelope to transfer any kinds of data. A message may contain an arbitrary number of named subsections which can hold any form of data. In one embodiment, the messages may be in a markup language. In one embodiment, the markup language is XML. Each peer's messaging layer may deliver an ordered sequence of bytes from the peer to another peer. The messaging layer may send information as a sequence of bytes in one atomic message unit. In one embodiment, messages may be sent between peer endpoints. In one embodiment, an endpoint may be defined as a logical destination (e.g. embodied as a URN) on any networking transport capable of sending and receiving Datagram-style messages. Endpoints are typically mapped into physical addresses by the messaging layer at runtime.

[0148] In one embodiment, a message may be a Datagram that may include an envelope and a stack of protocol headers with bodies and an optional trailer. The envelope may include, but is not limited to, a header, a message digest, (optionally) the source endpoint, and the destination endpoint. In one embodiment, each protocol header may include, but is not limited to, a tag naming the protocol in use and a body length. Each protocol body may be a variable length amount of bytes that is protocol tag dependent. Each protocol body may include, but is not limited to, one or more credentials used to identify the sender to the receiver. Such a message format preferably supports multiple transport standards. An optional trailer may include traces and accounting information. (Traversat, page 12, paragraphs 147-148)

While Traversat's peer-to-peer platform communicates messages that may include "an envelope and a stack of protocol headers with bodies and an optional trailer" (paragraph 148), at no point does Traversat describe that the messages have structured headers defined in accordance with an application messaging protocol. Rather, Traversat's messages may include "protocol headers" with information identifying one or more protocols, but those messages do not have a header structured in accordance with a protocol (e.g., in accordance with an application messaging protocol). Accordingly, Traversat fails to disclose or suggest at least the features of "defining an application message having a structured application message header, the structured message header being defined in accordance with an application messaging protocol, the structured message header comprising one or more components defined by the protocol with each of the one or more components relating to a corresponding set of attributes of the message," as required by applicant's independent claim 1.

Szabo describes a network connection management system (Abstract). The implementation of Szabo system includes communication of message between a Data Flow

Applicant(s): Frank Oliver Hoffmann, *et al.*
 U.S.S.N.: 10/816,445

Attorney Docket No.: 34874-096/2003P00878US

Segment (DFS) and at least one Control Segment of the system's controller (Szabo, paragraphs 47 and 73). With respect to the communicated messages, Szabo explains:

0073] The flow of messages from CS to DFS is asynchronous and independent of the flow of messages from the DFS to CS. Reply messages are initiated in response to query messages. The request and replies from the DFS and CS need not be interleaved. The querying segment should not be idle while waiting for a reply. The querying segment should not waste time trying to associate received replies with queries. Instead, reply messages should be self-contained so that the receiving segment will process the reply without needing to know what the original query was.

0074] Each message contains a serial number (or other indicator) that is generated by the originator of the flow. The originator of the flow is the first party to query or notify the other party regarding the flow. The serial number remains constant for all messages pertaining to the flow during the flow's lifetime. Since the DFS is typically the first party to detect an inbound flow (from the external network to the internal network), the DFS is generally the originator of the flow. In this case, the DFS sends a message (QUERY) to the CS to notify the CS about the new flow. In some instances (e.g. CS-assisted flow), the CS originates the flow by sending a message (NEWFLOW) to the DFS.

...

0076] FIG. 5 shows a table (table II) listing the data elements that are expected to be sent in a given message. Each message type may consist of a predefined subset of these data elements. The length of the message is derived from the UDP header.

0077] FIG. 6 shows a table (table III) of message fields for the message types defined in FIG. 4. After having read the present disclosure, it is understood and appreciated that other message types and message fields may also be utilized within the scope of this invention. The message layout is optimized for efficient processing by the CS and according to the needs of the particular DFS. Every message has a message header that includes msg_type, error_code, and message serial number fields. Example message layouts are shown in FIG. 9. (Emphasis added, Szabo, page 5, paragraphs 73-77)

Thus, while the messages used in Szabo's system include headers having, for example, msg_type, error_code and other fields, at no point does Szabo describe application messages that have headers structured in accordance to an application messaging protocol. Indeed, there is no indication in Szabo that the messages communicated are even defined by some given application. Rather, messages are presumably defined and/or generated by the DFS and/or the CS modules of the controller. Accordingly, Szabo fails to disclose or suggest at least the features of "defining an application message having a structured application message header, the structured message header being defined in accordance with an application messaging protocol, the structured message header comprising one or more components defined by the protocol with each of the one or more components relating to an associated set of attributes of the message," as required by applicant's independent claim 1.

Applicant(s): Frank Oliver Hoffmann, *et al.*
U.S.S.N.: 10/816,443

Attorney Docket No.: 34874-096/2003P008781US

With respect to Szabo's paragraph 120, relied upon by the examiner in rejecting independent claim 1, the paragraph states:

[0120] Although the processing is shown as sequential in FIG. 17, the controller continually receives packets in an asynchronous manner. For a segmented controller, the DFS continuously receives messages from the CS in an asynchronous manner. Since the controller is continually receiving packets and messages, the controller may not enter an idle state and continues processing messages and packets from a local memory buffer. (Szabo, page 8, paragraph 120)

The above-indicated paragraph merely indicates that the processing depicted in the flowchart of FIG. 17 corresponds to asynchronous receipt of packets. Contrary to the examiner's contentions, this paragraph does not "disclose processing mode for a message". This paragraph certainly does not disclose or suggest "defining an application message having a structured application message header, the structured message header being defined in accordance with an application messaging protocol, the structured message header comprising one or more components defined by the protocol with each of the one or more components relating to a corresponding set of attributes of the message".

Ringseth describes a system and methods for providing compile-time declarative modeling for SOAP-based data transmission, and the minimization of coding in connection with SOAP-based Web services by a developer (Ringseth, page 1, paragraph 3). With respect to SOAP-based communications, Ringseth explains:

[0051] A SOAP message is an XML document that comprises a SOAP envelope, an optional SOAP header and a SOAP body. The basic format of a SOAP message is illustrated in exemplary pseudocode 300 of FIG. 3A. The SOAP envelope is the top element of the XML document representing the message. The SOAP header is a generic mechanism for adding features to a SOAP message in a decentralized manner without prior agreement between the communicating parties. SOAP defines a few properties that can be used to indicate who should deal with a feature and whether it is optional or mandatory. The SOAP body is a container for mandatory information intended for the ultimate recipient of the message. For the envelope, the element name is "Envelope" and is present in the SOAP message. The element may contain namespace declarations as well as additional properties. If present, such additional properties are namespace-qualified. Similarly, the element may contain additional sub elements, but if present these elements are namespace-qualified and follow the SOAP body element. (Emphasis added, Ringseth, page 5, paragraph 51)

Thus, unlike applicant's messages which include a structured header defined in accordance with an application messaging protocol, Ringseth's messages include features added without prior agreement between the communicating parties. In other words, the features/properties/attributes of Ringseth's messages are not organized in a structured manner defined in accordance with some protocol. Accordingly, Ringseth also fails to disclose or

Applicant(s): Frank Oliver Hoffmann, *et al.*
U.S.S.N.: 10/816,445

Attorney Docket No.: 34874-096/2003P00878US

suggest at least the features of "defining an application message having a structured application message header, the structured message header being defined in accordance with an application messaging protocol, the structured message header comprising one or more components defined by the protocol with each of the one or more components relating to a corresponding set of attributes of the message," as required by applicant's independent claim 1.

Because none of the references relied upon by the examiner to reject the independent claims discloses or suggests, alone or in combination, at least the features "defining an application message having a structured application message header, the structured message header being defined in accordance with an application messaging protocol, the structured message header comprising one or more components defined by the protocol with each of the one or more components relating to a corresponding set of attributes of the message," applicant's independent claim 1 and the claims depending from it are patentable over the cited art.

Applicant's independent claims 5, 9, 10 and 11 recite "defining an application message having a structured application message header, the structured message header being defined in accordance with an application messaging protocol, the structured application message header comprising one or more components defined by the protocol with each of the one or more components relating to a corresponding set of attributes of the message", or similar language. For reasons similar to those provided with respect to independent claim 1, at least these features are not disclosed by the cited art. Applicant's independent claims 5, 9, 10 and 11, and the respective claims depending from them, are therefore patentable over the cited art.

CONCLUSION

It is believed that all of the pending claims have been addressed in this paper. However, failure to address a specific rejection, issue or comment, does not signify agreement with or concession of that rejection, issue or comment. In addition, because the arguments made above are not intended to be exhaustive, there may be reasons for patentability of any or all pending claims (or other claims) that have not been expressed. Finally, nothing in this paper should be construed as an intent to concede any issue with regard to any claim, except as specifically stated in this paper, and the amendment of any claim does not necessarily signify concession of unpatentability of the claim prior to its amendment.

On the basis of the foregoing amendments, applicant respectfully submits that the pending claims are in condition for allowance. If there are any questions regarding these

Applicant(s): Frank Oliver Hoffmann, *et al.*
U.S.S.N.: 10/816,445

Attorney Docket No.: 34874-096/2003P0087&US

amendments and remarks, the examiner is encouraged to contact the undersigned at the telephone number provided below.

The Commissioner is hereby authorized to charge any fees that may be due, or credit any overpayment of same, to Deposit Account No. 50-0311, Reference No. 34874-096.

Respectfully submitted,

Date:

Jan 2, 2008

Ido Rabinovitch

Ido Rabinovitch
Reg. No. L0080

MINTZ, LEVIN, COHN, FERRIS
GLOVSKY and POPEO, P.C.
Attorneys for Applicant
One Financial Center
Boston, Massachusetts 02111
Customer No. 64280
Telephone: 617/348-1806
Facsimile: 617/542-2241
email: irabinovitch@mintz.com

4222440v.1